We claim:

1) A multi-thread processor for processing a plurality of threads comprising:

a Thread ID generator producing a unique thread indication for each said thread;

a plurality of register sets, one said register set for each said thread, each said register set for each said thread comprising a plurality of registers;

an n-way register set controller coupled to said plurality of register sets and simultaneously handling multiple read or write requests for one or more of said unique threads;

a Fetch Address Stage for the generation of Program Memory Addresses;

a Program Access Stage for receiving Program Memory Data associated with said Program Memory Addresses;

a Decode Stage for converting said Program Memory Data into instructions, said Decode Stage coupled to said n-way register set controller;

a First Execution Stage for handling a multiply class of instruction received from said Decode Stage;

A Second Execution Stage for handling an Arithmetic Logical Unit class of instructions received from said Decode

Stage, said Second Execution Stage also coupled to said n-way register set controller;

A Memory Access Stage for handling reading and writing of external memory;

A Write Back Stage coupled to said n-way register set controller for writing data to said register set;

each said Stage performing an operation during a Stage Cycle;

said Thread ID value alternating from one stage cycle to the next.

2) The processor of claim 1 where a pipeline core is formed by stages in succession: said Fetch Address stage, said Program Access stage, said Decode stage, said First Execution stage, said Second Execution stage, said Memory Access stage, and said Write Back stage.

3) The processor of claim 1 where said n-way register set controller simultaneously receives at least one of read requests from said Decode stage, read and write requests from said Second Execution stage, or write requests from said Write Back stage.

4) The processor of claim 1 where said Memory Access stage is coupled to a memory controller.

5) The processor of claim 4 where said memory controller issues a stall signal when receiving a memory request to an external memory.

6) The processor of claim 4 where said memory controller issues a stall signal when receiving a memory read request to an external memory.

7) The processor of claim 4 where said memory controller issues a stall signal which lasts an interval from receiving a memory read request to receiving requested data form said external memory.

8) The processor of claim 1 where said pipeline core comprises a subset of said stages on one thread and remaining said stages on said other thread.

9) The processor of claim 1 where said first execution stage performs multiply operations and said second execution stage performs non-multiply instructions.

10) The processor of claim 1 where said decode stage forwards non-multiply operands to said second execution stage.

11) The processor of claim 1 where program memory contains a single instance of a program.

12) The processor of claim 1 where said thread ID can be read by each said thread.

13) The processor of claim 1 where each said thread reads said thread ID to perform thread operations which are independent.

14) The processor of claim 1 where thread ID is used along with an address to decode a device in a memory map.

15) The processor of claim 1 where devices are decoded in a memory map based on address only.

16) The processor of claim 1 where said Decode stage performs decoding of instructions for said multiply class of instruction, and said First Execution stage performs decoding of instructions for said arithmetic logical unit class of instructions.

17) The processor of claim 16 where if one of said multiply class of instructions requires a register operand,

said operand is provided from said registers to said decode stage.

18) The processor of claim 16 where if one of said arithmetic logical unit class of instructions requires a register operand, said operand is provided from said registers to said first execution stage.

19) The processor of claim 1 where at least one said stage includes an operational clock which is at a higher rate than said stage clock.

20) A multi-threaded processor comprising a plurality of stages operating on a stage clock for passing information from stage to stage, each stage including inter-stage storage for thread information associated with a thread ID;

a first said stage receiving program counter address information from a unique program counter for each said thread ID and delivering said address to a program memory;

a second stage for receiving program data from a program memory;

a third stage for performing decode of said program data;

a fourth stage for performing multiplication operations or decode operations;

a fifth stage for performing non-multiplication operations;

a sixth stage for accessing external memory;

a seventh stage for writing results of computations performed in said fourth stage or said fifth stage back to a register set;

said register set being duplicated for each said thread ID;

each said first through seventh stage receiving said thread ID and operating according to a first or second value.

21) The multi-threaded processor of claim 20 where said first, third, fifth and seventh stages use one value for said thread ID, and said second, fourth, and sixth stages use a different value for said thread ID.

22) The multi-threaded processor of claim 20 where said threads each control execution of a program, and said programs execute independently of each other.

23) The multi-threaded processor of claim 22 where one said thread may stop execution and the other said thread continues execution.

24) The multi-threaded processor of claim 20 where said registers and said stages contain data which is used separately for each said thread ID.

25) The multi-threaded processor of claim 20 where said stages alternate between two threads on each said stage clock.

26) The multi-threaded processor of claim 20 where said thread-ID identifies a register set and a program counter.

27) The multi-threaded processor of claim 20 where said third stage performs said decode for multiply operations.

28) The multi-threaded processor of claim 20 where said fourth stage performs said decode for non-multiply operations.

29) The multi-threaded processor of claim 27 where said fourth stage performs said multiply operations.

30) The multi-thread processor of claim 28 where said fifth stage performs said non-multiply operations.

31) The multi-thread processor of claim 28 where said non-multiply operations include at least one of rotate, shift, add, subtract, or load.

32) The multi-thread processor of claim 29 where said multiply operations include multiplication by a constant from one of said registers.

33) The multi-thread processor of claim 30 where said non-multiply operations include addition of a multiply result from said fourth stage.

34) The multi-thread processor of claim 20 where said thread ID includes a plurality of values, each said value having at least one register and a program counter.

35) The multi-thread processor of claim 20 where said sixth stage said external memory responds in more than one said stage clock cycle.

36) The multi-thread processor of claim 20 where said external memory generates a stall signal for each said thread ID, thereby causing all said stages to store and maintain data for that thread ID until said stall signal is removed by said external memory.

37) The multi-thread processor of claim 20 where said fifth stage generates an address for a data memory.

38) The multi-thread processor of claim 37 where said sixth stage receives and generates data for said data memory.

39) The multi-thread processor of claim 20 were said thread information storage includes registers which store results from said fifth stage for each said thread ID.

40) The multi-thread processor of claim 20 where said registers which store results from said fifth stage allow a non-stalled thread to continue execution without modifying said stored results.